Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

ELSEVIER

Computer Networks 41 (2003) 743–760

COMPUTER NETWORKS

www.elsevier.com/locate/comnet

Anycasting-based protocol for geocast service in mobile ad hoc networks ☆

Young-Bae Ko [a,*], Nitin H. Vaidya [b]

[a] *Department of Information and Computer Engineering, Ajou University, Suwon, Republic of Korea*
[b] *Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA*

Received 4 January 2002; received in revised form 17 July 2002; accepted 21 October 2002

Responsible Editor: R. Sivakumar

Abstract

Geocasting is a variation on the notion of multicasting. Geocasting is useful for sending messages to nodes in a specified geographical region. This region is called the geocast region. This paper presents a protocol, named *GeoTORA*, for geocasting in mobile ad hoc networks. The proposed GeoTORA protocol combines anycasting with local flooding to implement geocasting. Thus, GeoTORA requires two phases for geocasting. First, it performs *anycasting* from a source to any node in the geocast region, by modifying the Temporally-Ordered Routing Algorithm (TORA) (unicast) routing protocol. Subsequently, localized flooding within the geocast region is performed to deliver the messages to nodes within the geocast region. This integration of TORA and local flooding can significantly reduce the overhead of geocast delivery, while maintaining reasonably high accuracy.
© 2002 Elsevier Science B.V. All rights reserved.

1. Introduction

This paper considers the problem of providing a geocast service in mobile ad hoc networks. *Geocasting* is a mechanism to deliver messages of interest to all nodes within a given geographical region [18]. There can be many useful services provided using a geocast. Sending emergency messages to a specific area or delivering geographic-oriented advertisements are examples of such geocast services.

Geocasting is similar to traditional multicasting in that it involves more than one destination. However, there is a significant difference between the two approaches. In traditional multicasting, a host becomes a member of the multicast group by explicitly joining the multicast group, and, potentially, any node may join a multicast group. On the other hand, in geocasting, a host automatically becomes a member of a geocast group if (and only if) its location belongs to the region associated with the geocast group—this region is referred to as the *geocast region* [21]. Thus, the set of nodes in the geocast region is said to form the *geocast group*. For a node to be able to determine whether

☆ This paper was presented in part at the 9th International Conference on Networking Protocols, 2000.

* Corresponding author.
E-mail addresses: youngko@ajou.ac.kr (Y.-B. Ko), nhv@uiuc.edu (N.H. Vaidya).
URL: http://www.crhc.uiuc.edu/nhv.

1389-1286/03/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
doi:10.1016/S1389-1286(02)00437-1

it belongs to a geocast group or not, the node needs to be location-aware (that is, should be able to know its own physical location). A node can determine its physical position in a number of different ways (see [16], for more information), for instance, using the global positioning system (GPS) or local positioning systems [6,7].

The proposed protocol in this paper is named "GeoTORA", because it is derived from the Temporally-Ordered Routing Algorithm (TORA) (unicast) routing protocol [28]. TORA is first modified to be able to perform *anycast* and our GeoTORA protocol is then obtained using a small variation on the anycasting protocol. Flooding is also incorporated in GeoTORA, but it is limited to nodes within a small region. This integration of TORA and local flooding can significantly reduce the overhead of geocast delivery, while maintaining high accuracy, as shown in this paper.

This paper is organized as follows. Section 2 summarizes related work. Section 3 outlines a general description of GeoTORA, without presenting any implementation details. A more detailed description of TORA and GeoTORA is presented in Sections 4 and 5. Performance evaluation results are analyzed in Section 6. Finally, Section 7 presents our conclusions.

2. Related work

This paper presents a new algorithm for geocasting in mobile ad hoc networks. Broadly defined, a mobile ad hoc network is a network formed without any central administration or preexisting infrastructure, consisting of only mobile hosts that communicate with each other over wireless links [10]. In a mobile ad hoc network, typically, all mobile hosts behave as routers and a route between a pair of nodes may go through several other mobile nodes. These routes can change when hosts [1] change location. Therefore, there has been significant research on the development of (unicast) routing protocols for mobile ad hoc networks [2,5,9,15,19,20,22,28,30,31,34].

[1] We will use the terms *node* and *host* interchangeably.

In addition to the above work on unicast routing in mobile ad hoc networks, there has been significant work on multicasting as well, and several approaches have been proposed [4,8,14,17,23,26,32,33,35,36]. The schemes for multicasting can be broadly divided into two types: *flooding*-based schemes and *tree*-based schemes. Both approaches have their advantages and disadvantages. Flooding-based schemes do not need to maintain as much network state as the tree-based protocols. On the other hand, flooding-based schemes can potentially deliver the multicast packets to a large number of nodes who do not wish to receive them (i.e., nodes which do not belong to the multicast group). Tree-based schemes tend to avoid this drawback of flooding-based schemes, at the cost of increased overhead in tree maintenance.

The concept of geocasting was introduced by Imielinski and Navas [18]. They also presented an architecture to implement geocasting in the Internet. We have previously considered [21] the use of geocasting in mobile ad hoc networks, and developed the location-based multicast (LBM) algorithm that uses flooding to deliver a geocast packet. To reduce the propagation of the flood, LBM limits the flood to a *forwarding zone*—the forwarding zone covers a subset of the network, and is determined based on the location of the sender and coordinates of the geocast region. Although the algorithm in [21] is able to limit the flood of geocast packets to a relatively small region, still many nodes outside the geocast region tend to receive the geocast packet. This paper basically improves on the algorithm presented in [21]. Another approach similar to [21] but based on a mesh topology has been proposed in [3].

Another protocol for geocasting in mobile ad hoc networks, named *GeoGRID* [24] has been proposed as an extension of the GRID unicast routing protocol [25]. In this protocol, a network is logically partitioned into a number of square shaped regions and one node is elected as the leader (i.e., the gateway) of each square. Geocast message delivery is then performed by those gateway nodes only in a grid-by-grid manner, resulting in possibly better performance than LBM. While GeoGRID and the proposed GeoTORA share the common objective of improving the LBM protocol, they are inherently different approaches having their advantages and disadvantages with different scenarios. Thus, GeoGRID (and [3], as well) belong to the flooding-based approach, whereas GeoTORA belongs to the graph-based or tree-based approach.

Ref. [1] addresses the problem of supporting geocast services for cellular mobile networks and presents a cellular architecture for geocasting.

3. An overview of GeoTORA

This section is intended to provide a simplified description of the GeoTORA protocol and insight into its operation. Since GeoTORA is based on TORA unicast routing protocol [28], we begin with an abstract description of TORA.

3.1. TORA and link reversal algorithms

TORA is one of a family of *link reversal* algorithms [13] for routing in ad hoc networks. For *each* possible destination in the ad hoc network, TORA maintains a *destination-oriented* directed acyclic graph (DAG). In this graph structure, starting from any node, if links are followed in the logical direction of the links, the path leads to the intended destination. TORA uses the notion of *heights* to determine the direction of each link—we will discuss this in greater detail later. Despite dynamic link failures, TORA attempts to maintain the destination-oriented DAG such that each node can reach the destination, as illustrated below.

Fig. 1 illustrates how link reversal is performed in TORA. An arrow connecting a pair of nodes in this figure implies that the two nodes can communicate with each other. That is, the physical link between the two nodes is bidirectional. However, the TORA algorithm imposes a *logical* direction on the links, as illustrated in Fig. 1(a)—this figure shows the destination-oriented DAG with node G being the destination. Observe that, starting from any node in the graph, the destination G can be reached by simply following the directed links.

Now assume that the link between nodes D and F breaks (perhaps because node F moves away from node D). Then, in the destination-oriented DAG, node D does not have any outgoing logical link. In response, TORA *reverses* logical direction of the (D,B) and (D,C) links, as shown in Fig. 1(b). Now, node C does not have any outgoing logical link. In response, logical direction of link (B,C) is reversed, [2] resulting in the graph in Fig. 1(c). Now since node B does not have any outgoing logical link, the logical direction of link (A,B) is reversed, resulting in the destination-oriented DAG in Fig. 1(d). In this state, each node (other than the destination G) has an outgoing logical link, and is able to reach the destination node G by following the directed links.

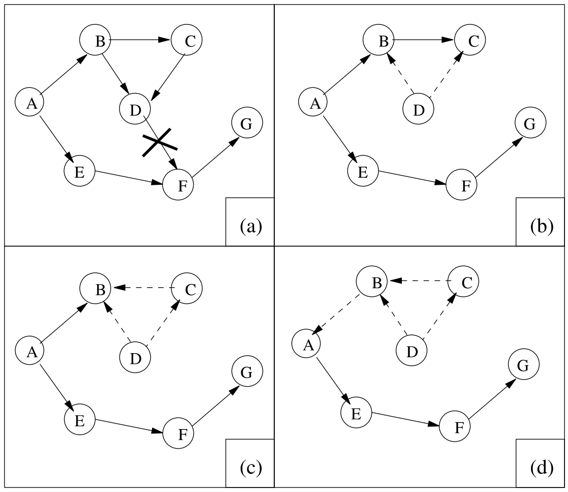


Fig. 1. An example of link reversal algorithm in route maintenance phase of TORA: Route maintenance is required due to failure of the link between nodes D and F.

3.2. Anycasting using modified TORA

To implement GeoTORA, we first modify TORA to be able to perform *anycast* [12,29]. To perform an *anycast*, an anycast group is defined—anycast group consists of a subset of the nodes in the network. When a node sends a message to the anycast group, the message is delivered to any one member of the anycast group.

[2] TORA employs a partial reversal algorithm. Thus, only some or all incoming links at a node may be reversed.

While TORA maintains a DAG for each destination, the anycasting algorithm would maintain a *single* DAG for a given *anycast group*. Observe that, in steady state, when using TORA, only the intended destination node is a *sink* node in its destination-oriented DAG. To perform anycasting, we modify TORA to maintain a DAG structure such that all nodes belonging to the anycast group are sinks. In this case, a link that is between two nodes belonging to anycast group is not given a logical direction.

Fig. 2 illustrates the anycast scheme. In this case, let us assume that nodes A–D belong to the anycast group. The present DAG structure is shown in Fig. 2(a). Observe that links with both endpoints in the set {A,B,C,D} do not have any logical direction. From any node that is outside the anycast group, following the directed links leads to one member of the anycast group. Now, suppose that node G moves, breaking link (A,G)—the resulting DAG structure is shown in Fig. 2(b). Observe that now node G does not have any outgoing link. In response, the logical direction of link (G,J) is reversed, resulting in the DAG shown in Fig. 2(c). Now all nodes that are outside the anycast group have an outgoing link (and a path to at least one node in the anycast group).

3.3. GeoTORA protocol

The GeoTORA protocol is obtained using a small variation on the above anycasting protocol. Consider the system shown in Fig. 3(a). In this case, let us assume that the circle represents the

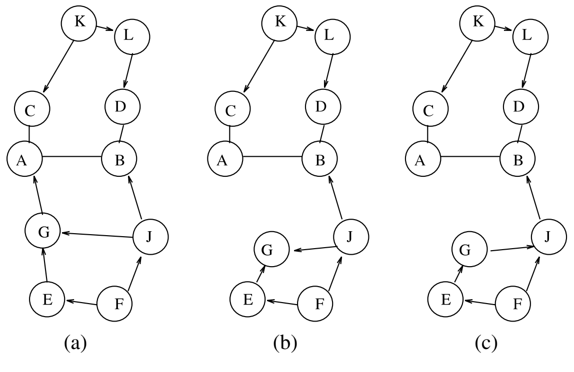


Fig. 2. Anycasting using modified TORA.

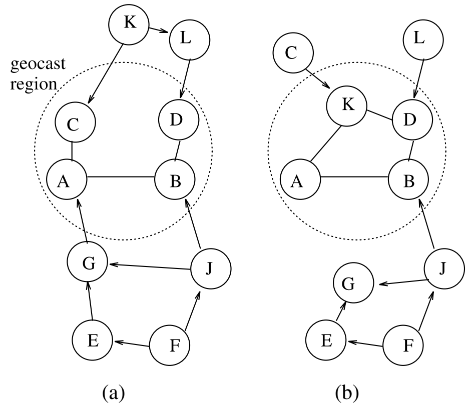


Fig. 3. Geocasting using GeoTORA.

geocast region. Thus, the geocast group at the present time is the set of nodes {A,B,C,D}. GeoTORA maintains a single DAG for each geocast group—the DAG is updated when membership of the geocast group changes.

To perform geocasting using GeoTORA, first, a sender node essentially performs an *anycast* to the geocast group members—similar to the above anycast protocol, in GeoTORA, logical directed links are set up such that a node wishing to perform a geocast can reach any one node in the geocast group by simply forwarding the packet on any outgoing link. When any node in the geocast group receives the packet, it *floods* the packet such that the flooding is *limited to the geocast region*—to achieve this, only nodes that are within the geocast region (i.e., the geocast group members) forward the flooded packet; other nodes simply drop the flooded packet. To ensure that a given node does not forward a flooded packet more than once, a sequence number is attached to each packet, similar to the flooding schemes used in other protocols [20,22].

For instance, if node E in Fig. 3(a) wants to perform a geocast, it forwards the geocast packet to node G, along the outgoing link (E,G). Node G, in turn, forwards the packet to node A. Since node A is in the geocast region, it initiates a flooding of the packet limited to the geocast region. Nodes B and C, on receiving the packet from node A, forward the packet to their neighbors. When node A receives the packet from node B or C, it does not forward the packet, since node A has already once forwarded the packet to its neighbors. In this manner, the packet will reach nodes A, B, C and D that belong to the geocast region.

Since mobile hosts may move into and out of the geocast region, the set of nodes in the geocast group can change dynamically. Thus, we need to incorporate mechanisms to allow a node that is not a *sink* (i.e., a group member) to become a sink, and vice versa. GeoTORA incorporates such mechanisms, as described in more detail in the next section. Here we explain the behavior of the protocol by continuing with our example.

Again, consider Fig. 3(a). Now suppose that node C moves out of the geocast region, and node K moves into the geocast region simultaneously. The resulting DAG (after GeoTORA takes appropriate steps) is shown in Fig. 3(b). Observe that node K has now become a sink, and node C is no more a sink. Since node K has moved closer to node A, a link exists between nodes A and K. However, this link is not given a logical direction, since nodes A and K are now both in the geocast region.

There are two other possibilities that need to be handled in GeoTORA: (a) GeoTORA needs to handle the case when all the geocast members may be partitioned from some of the other nodes in the network. (b) GeoTORA also needs to handle the case when the geocast region is empty. In this case, eventually, some node may again enter the geocast region. Thus, the transitions between a non-empty geocast group and an empty geocast group must be considered.

4. Temporally-Ordered Routing Algorithm [28]

Since GeoTORA is based on TORA, we now present some more details of TORA. [3] Readers familiar with the details of TORA may omit this section without loss of continuity. TORA provides loop-free, (potentially) multiple routes from any source to a desired destination. In order to forward data packets to a given destination, a node simply needs to maintain direction for its links. Logical direction of a link between a pair of nodes is determined by assigning a *height* to each node. The logical links are considered to be directed from nodes with higher height towards nodes with lower height—lexicographic ordering on height is used since height is defined as a five-tuple, (τ, oid, r, δ, i), as explained below. The height consists of two components: a *reference level* represented by the first three components of the five-tuple, and a *delta* with respect to the reference level, represented by the last two components of the five-tuple. Each component of the five-tuple is explained below:

- τ: a new reference level is defined each time a node loses its last outgoing link. τ is a tag that represents the time of the link failure.
- *oid* (originator id): Unique identifier of the node that defined the new reference level. The *oid* ensures that the reference levels can be totally ordered lexicographically even if multiple nodes define reference levels simultaneously.
- r: reflection indicator bit. This bit is initially set to 0. As seen earlier, when a node does not have any outgoing links as a result of a link failure, it reverses some (or all) of its incoming links. The reaction to a link failure propagates through other nodes that have lost all their routes to the destination, as a result of the link failure. When the reaction propagates to a node that originally had only outgoing links, but now has no outgoing links (since all the outgoing links were reversed by its neighbors), the node "reflects" the link reversals, by setting its height higher than any of its neighbors. The r bit is used for this purpose.
- δ: propagation ordering parameter. The use of this parameter will be explained by means of an example below.
- i: unique node identifier (ID).

TORA performs three basic functions: route creation, route maintenance, and route erasure. Three control packets—query (QRY), update (UPD), and clear (CLR)—are used to accomplish these functions. Creating routes from various sources to the destination corresponds to establishing a sequence of directed links from each

[3] Due to lack of space, it is not possible to illustrate all details of TORA with sufficient clarity. The readers are referred to [28] for further explanations.

source to the destination. This is accomplished by maintaining a DAG rooted at the destination. A query/reply process with QRY and UPD packets is used for building the destination-oriented DAG. Fig. 4 illustrates the process of route creation, with time increasing from Fig. 4(a)–(f).

Initially, as shown in Fig. 4(a), height H_i of each node i (other than the destination) is set to NULL—specifically, $H_i = (-,-,-,-,i)$. Note that although the last component in H_i is not null, the height is considered to be equal to NULL. The destination node G sets its height to be ZERO $= (0,0,0,0,G)$. Now, when any node with no outgoing links (for instance, node A in Fig. 4(a)) requires a route to the destination (node G in Fig. 4), it broadcasts a QRY packet to all of its neighbors and sets a *route-required* flag. When a node, say X, receives a QRY packet, it reacts in accordance with the following rules:

(a) If node X has no downstream links and its route-required flag is un-set, then it just forwards the QRY to neighbors, while setting the route-required flag. Note that a link between two nodes whose height is NULL is considered *undirected*. On the other hand, a NULL height is considered to be higher than any non-NULL height.
Observe that nodes B and E in Fig. 4(b) apply this rule on receiving a QRY from node A, and forward the QRY packet to their neighbors. In Fig. 4, a double circle around a node indicates that the route-required flag is set at that node.

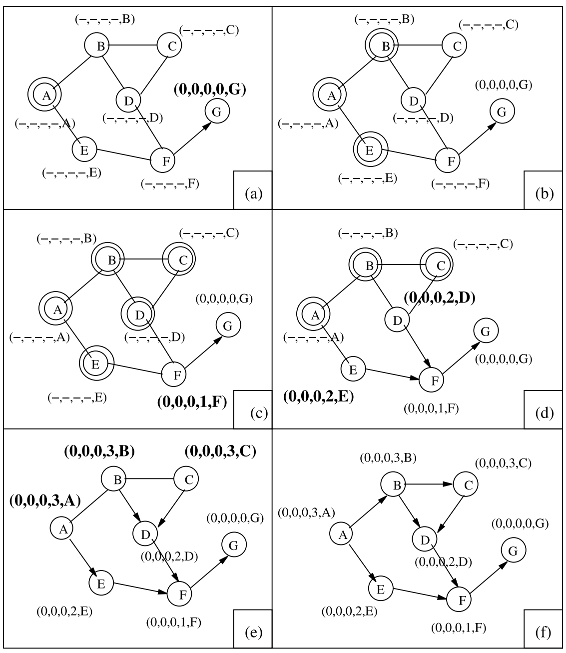


Fig. 4. Route creation phase in TORA: In the figure, a circle around a node indicates that the route-required flag is set. Arrows on each wireless link points from the higher height node to the lower height node. The height is depicted as a five-tuple, as explained in the context.

(b) If node X has no downstream links, but its route-required flag is set, then node X simply discards the received QRY packet. For example, when node A receives a QRY from nodes B or E, it will drop the QRY without any further reactions (since node A's route-required flag was set when it forwarded the query to its neighbors).

(c) If node X has at least one downstream link and its height is NULL, it modifies value of δ in its height, based on the relative height metric of neighboring nodes. Thus, node X changes its current NULL height $(-,-,-,-,\mathrm{X})$ to $(\tau, oid, r, \delta+1, \mathrm{X})$, where $(\tau, oid, r, \delta, i)$ is the minimum height of its non-NULL neighbors (this height corresponds to some neighbor node, i). Also, node X sends an UPD packet containing the new height to its neighbors.
In our example, in Fig. 4(c), node F updates its height to $(0,0,0,1,\mathrm{F})$, since the only non-NULL height among its neighbors is height $(0,0,0,0,\mathrm{G})$ of the destination node G. Node F, then, transmits an UPD to its neighbors.

(d) If node X has at least one downstream link, but its height is non-NULL, it first compares the time the last UPD packet was broadcasted with the time when the link over which the QRY packet was received became active. If an UPD packet has been broadcasted since the link became active, it simply discards the QRY; otherwise, node X broadcasts an UPD packet as a response, to inform its height to neighbors.
For instance, in Fig. 4(d), node F may receive a QRY packet from node D after node F has sent an UPD packet, as seen earlier (with reference to Fig. 4(c)). This results in node F discarding the QRY packet from node D.

When a node, say Y, gets an UPD packet from its neighbor, node Y checks its route-required flag to see if it is set or not. If the flag is set (meaning that the height of node Y is NULL), then it updates its height as $(\tau, oid, r, \delta + 1, \mathrm{Y})$, based on the minimum height value $(\tau, oid, r, \delta, i)$ of its non-NULL neighbors. Node Y then broadcasts an UPD containing its new height. On the other hand, if the route-required flag of node Y is unset, Y only reacts if it has lost its last downstream link. As an example in Fig. 4(e), nodes B and C update their heights in this manner. In turn, node A updates its height as $(0,0,0,3,\mathrm{A})$ since its route-required flag was set and its non-NULL neighbors' minimum height is $(0,0,0,2,\mathrm{E})$ when it receives an UPD from node E. When route creation process initiated by node A in Fig. 4 completes, the heights of the nodes along any route to the destination are strictly decreasing, as shown in Fig. 4(f).

Destination-oriented DAG established by the route creation process can break due to a link failure. In this case, a procedure for maintaining routes is necessary in order to rebuild the DAG rooted at the destination. TORA does not react to link failures as long as there are still outgoing links available at each node (other than the destination). If some node, say node Z, loses all its outgoing links, then it reverses the direction of some or all of its incoming links. Link reversal is performed by choosing a new *reference level* for the height such that the height of node Z becomes higher than any other node in the network. The node that chooses a new reference level then broadcasts an UPD packet containing the new reference level to its neighbors. If such a link reversal by node Z causes another node (say, node W) to lose its last downstream link, node W adjusts its height to be "lower" than the height of the sender of the UPD, i.e., node Z, and broadcasts an UPD. This process of link reversal [4] continues until either all nodes have at least one downstream link (see Fig. 1) or a network partition is detected.

[4] Remind that TORA is based on a partial link reversal algorithm. As an example, in Fig. 1(c), node C only reverses the link from node B to node C, but not the link from node D to node C.

One of advantages in TORA protocol is that a network partition can be detected during the route maintenance phase. This capability leads to a procedure for erasing routes. In the route erasure process, a CLR control packet is flooded throughout the network to erase all invalid routes so that all links of nodes partitioned from the destination become undirected.

5. Proposed GeoTORA protocol

We now further elaborate on GeoTORA. Since GeoTORA is quite similar to TORA, we primarily highlight the differences between TORA and GeoTORA in this section. First, the route creation and maintenance in GeoTORA is discussed, followed by the procedure for delivery of geocast messages using GeoTORA. Recall that, as discussed in Section 3, for each geocast group, GeoTORA maintains a single DAG. This is similar to the DAG maintained by TORA, with the difference being that *all nodes* that belong to the geocast region have a ZERO height—link between a pair of nodes is not assigned a direction if both nodes have ZERO height. This is unlike TORA, where only a single node (the destination) has ZERO height.

5.1. Route creation and maintenance in GeoTORA

In order to deliver packets to the geocast group, a source should have a route to the given geocast region. To establish routes initially, GeoTORA uses a *route creation process* that is essentially identical to that for TORA, but with the difference noted above (i.e., all geocast members have ZERO height). Fig. 5 provides an illustration for the process of geocast route creation in GeoTORA.

In Fig. 5, the dotted circle represents the geocast region—nodes G, H and I are within the geocast region (in this example, the set of nodes in the geocast region does not change). Fig. 5(a) represents the initial state of the system. Since nodes G, H and I are within the geocast region, they set their height to ZERO. Any other node, say i, sets its height to NULL—specifically, node i sets its height to be $(-,-,-,-,i)$. Note that links between two nodes with ZERO height are not assigned any

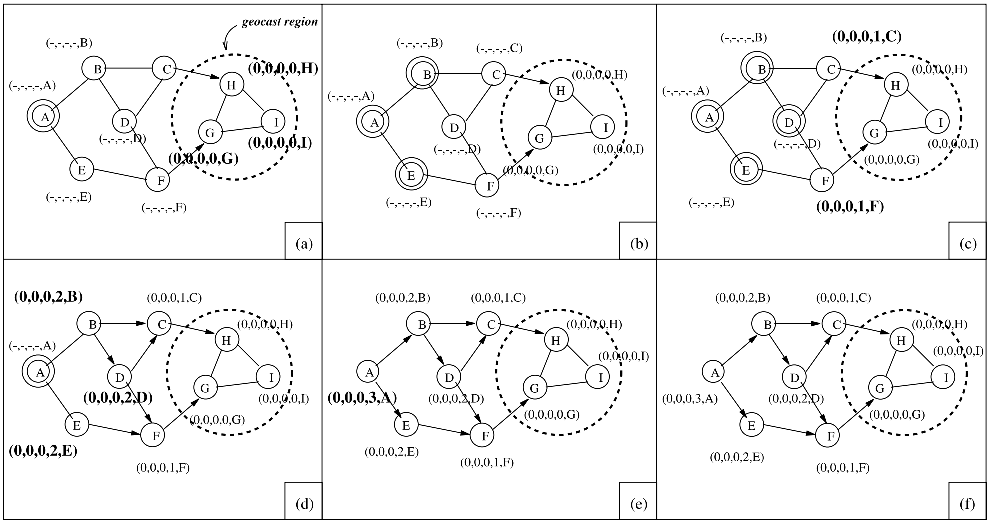


Fig. 5. Geocast route creation in GeoTORA.

direction; similarly links between two nodes with NULL height are also not assigned any direction. Nodes C and F (whose height is NULL) have links, respectively, to nodes H and G (whose height is ZERO). Therefore, the links (C,H) and (F,G) are assigned a direction—recall that NULL height is considered to be greater than any non-NULL height. Rules for route creation process in GeoTORA are identical to those described for TORA.

Assume that node A wishes to perform a geocast to the geocast group. Since node A does not have any outgoing link, it transmits a QRY packet to its neighbors, and sets its *route-required* flag. Note that in Fig. 5, a double circle around a node indicates that the route-required flag at that node is set. The QRY packet transmitted by node A reaches nodes B and E, and they, in turn, forward the packets to their neighbors, and also set the local route-required flag (refer Fig. 5(b)). Nodes C and D receive the QRY message from node B, and node F receives from node E. In Fig. 5(c), observe that nodes C and F have outgoing links to geocast group members, but node D does not. Therefore, only node D forwards the packets to its neighbors, and sets its route-required flag. On the other hand, on receiving a QRY, nodes C and F change their height from NULL to $(0, 0, 0, 1, \mathrm{C})$ and $(0, 0, 0, 1, \mathrm{F})$, respectively, and send UPD message to their neighbors informing the new height. Response of a node on receiving an UPD message is identical to that in TORA. Fig. 5(d)–(f) show evolution of the algorithm beyond the stage shown in Fig. 5(c). At the end, as seen in Fig. 5(f), a DAG is established wherein each geocast group member is a sink.

Now we illustrate *route maintenance* in GeoTORA. In GeoTORA, the DAG may need to be modified when: (a) a link failure occurs, or (b) when a node enters or leaves the geocast region.

The GeoTORA route maintenance procedure in response to link failures is similar to TORA. Fig. 6 illustrates how the DAG is modified in GeoTORA in response to link failures, considering several link failure scenarios. Fig. 6(a) shows the case where no maintenance reaction is taken by node D, as a result of breakage of link (D,F), since node D still has an outgoing link (D,C). Next, as shown in Fig. 6(b), the link from node C to node H breaks. Now, node C is left without any outgoing links—let us assume that the link failure occurred at time 1. Node C then updates its height using a new reference level representing the fact that node C has lost all downstream links at time 1. The new height of node C is $(1, \mathrm{C}, 0, 0, \mathrm{C})$, as shown in Fig. 6(c). Node C also generates an UPD containing its new height and broadcasts the UPD to neighbor

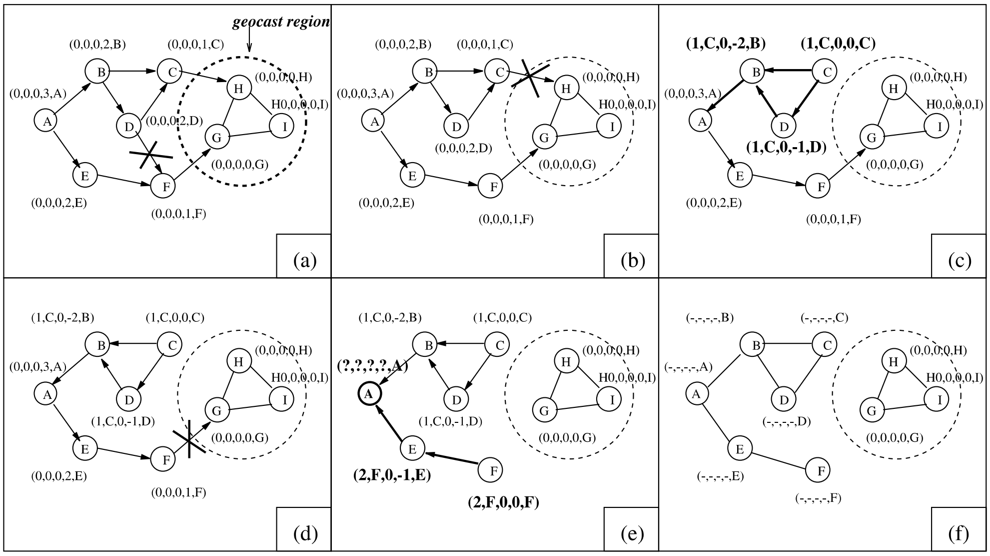


Fig. 6. Geocast route maintenance in GeoTORA: Three different scenarios of link failure.

nodes—the procedure for handling the UPD messages in GeoTORA is identical to TORA. Since node C increases its height, now node D also has no outgoing links—in response, node D chooses height $(1, C, 0, -1, D)$ and sends an UPD to its neighbors. The new height chosen by nodes C and D results in the loss of the only outgoing link from node B. Therefore, node B lowers its height to $(1, C, 0, -2, B)$, to be lower than the current height of node D, and transmits an UPD containing its new height. The new height chosen by node B again causes reversal of the link between nodes A and B to, now, point to node A. However, node A still has another outgoing link (A,E), so no further action is needed. The final state of the DAG, after the failure of the link between nodes C and H, is shown in Fig. 6(c).

Sometimes, a link failure causes a network partition, such that some nodes may not have any path remaining to any node in the geocast group. For instance, Fig. 6(d) depicts the case where link between nodes F and G is broken. Now, assume that the time when failure occurred is 2. The reaction to this link failure is similar to the reaction following failure of link (C,H) in Fig. 6(b). As a result of the failure of link (F,G), nodes F and E choose new height. The resulting state is shown in Fig. 6(e). Observe that, before all the link failures, node A only had outgoing links (i.e., no incoming links in Fig. 6(a)). Now, however, due to the new height chosen by node E in Fig. 6(e), node A has no outgoing links remaining. Node A realizes that all its outgoing links are broken when it receives an UPD message from node E containing node E's new height. Subsequently, following the "reflection" procedure as defined in TORA, the fact that A is partitioned from the geocast group is detected. Therefore, *route erasure phase* is initiated. Details of the route erasure phase are not illustrated here for brevity—however, note that the procedure is identical to TORA. Fig. 6(f) shows the network state after route erasure process has been completed. Until a new route to the geocast region is detected, a source that is partitioned from the geocast group is not able to send geocast data packets.

Fig. 7 illustrates how GeoTORA handles geocast group membership changes. Consider Fig. 7(a) as an example network. In Fig. 7(a), when node C moves into the geocast region and becomes a sink, it simply updates its current height to be ZERO, and then a UPD is transmitted by node C to inform its new height to its neighbors. The resulting state is shown in Fig. 7(b). Now, let us assume that node H leaves the geocast group by

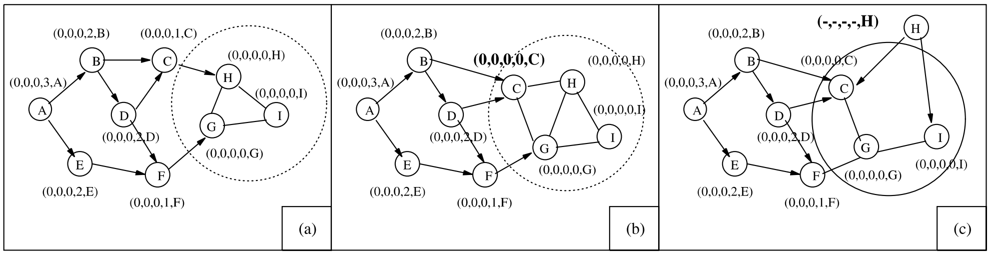


Fig. 7. Route maintenance in GeoTORA for handling dynamic change of geocast group.

moving out of the geocast region. In this case, the height of node H is set to NULL. Note that a NULL height is considered greater than any non-NULL height. Therefore, undirected links (H,C) and (H,I) in Fig. 7(b) now have logical directions from node H to C and from H to I, respectively, as shown in Fig. 7(c).

5.2. Delivery of Geocast packets

Geocast delivery using GeoTORA consists of two phases: *anycasting* phase and *local flooding* phase, as discussed below.

Anycasting phase: When a node wishes to send a packet to the geocast group, it forwards the packet on any of its outgoing links. [5] Each node that receives the packet forwards the packet on an outgoing link. Provided the source node is not partitioned from the geocast group, the packet eventually reaches *one member* of the geocast group.

Local Flooding Phase: Once a packet is delivered to one node in the geocast group (by the anycasting phase above), that node initiates *local flooding* of the packet. The purpose of local flooding, described below, is to deliver the packet to the remaining geocast group members. The node, say X, that initiates the flood, tags the specification of the *geocast region* to the packet, and broadcasts it to its neighbors. Any node, say Y, that receives the flooded packet verifies whether it is within the region whose specification is tagged to the packet. If node Y is outside the region, then it simply discards the packet. On the other hand, if node Y is within the tagged region, then node Y broadcasts the packet to its neighbors. Caution is taken to ensure that a given node would not broadcast the same packet more than once.

Local flooding initiated by node Y may not necessarily deliver the packet to *all* nodes within the *geocast region*. Particularly, using the above procedure, the packet would not be delivered to a geocast group member, say Z, if there is no path from Y to Z that consists of nodes belonging to the geocast region only (since local flooding above is confined to the geocast region). The probability that the packet would be delivered to all the geocast group members can be increased by using a larger region for the local flooding. There exists a trade-off between the overhead of local flooding and the number of geocast group members who receive the packet.

6. Performance evaluation

For the evaluation purpose, the proposed GeoTORA protocol is compared to *pure geocast flooding* and the LBM [6] [21] algorithms. Pure geocast flooding floods the whole network, and LBM scheme limits the flooding to the smallest rectangular region containing a source node and a geocast region. We performed a simulation study using an extended version of the network simula-

[5] If no outgoing link is available, then the appropriate steps in route creation and maintenance procedures are first invoked.

[6] Ref. [21] presents two LBM algorithms. We compare with their first algorithm which is based on flooding in a small region.

tor *ns*-2 [5]. *ns*-2 is a discrete event-driven network simulator with extensive support for simulation of TCP, routing, and multicast protocols. The extensions implemented by CMU Monarch Project were used for our simulations. Their extensions enable simulation of multi-hop wireless ad hoc networks. Extensions include simulation modules for the IEEE 802.11 MAC layer protocol and a radio propagation model.

6.1. Simulation model

In our simulation model, initial locations (X and Y coordinates) of the nodes are obtained using a uniform distribution. The nodes, chosen to be 30 nodes, move around in a rectangular region of size 700 unit $\times$ 700 unit square according to the following mobility model: each node chooses a direction, moving speed, and distance of move based on a predefined distribution and then computes its next position P and the time instant T of reaching that position. Each node moves with three different *maximum* speeds: 5, 10 and 20 units/s (i.e., average speeds of 2.5, 5 and 10 units/s, respectively). We ran our simulations with movement patterns generated for several different pause times, from 0 to 1000 s. A pause time of 0 s corresponds to continuous motion, whereas a pause time of 1000 s is equivalent to static networks, i.e., zero mobility, since our total simulation time is 1000 s.

Two mobile hosts are considered disconnected if they are outside each other's transmission range, which is defined as 250 units for all nodes. The wireless link bandwidth is 2 Mbps. One of the nodes is chosen as the sender for the geocasts—it initiates a geocast.

For the simulations, any data packets that cannot be delivered due to a broken route are simply dropped. The size of data payload is 512 bytes. Unless otherwise stated, 1000 geocasts have been done in each simulation run. For GeoTORA simulation, control packets are required to maintain the DAG, and the size of those packets is 32 bytes. Finally, a geocast region is defined to be a 200 unit $\times$ 200 unit square region with both X and Y coordinates in the range between 500 and 700.

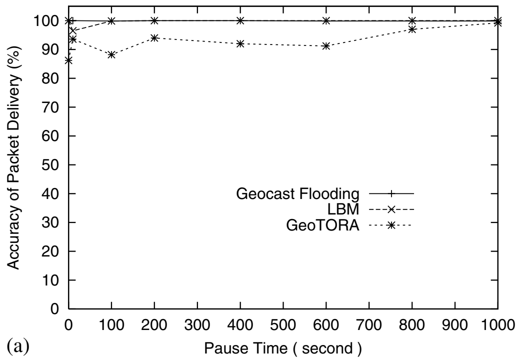
We use two performance metrics to measure the *accuracy* and *overhead* of geocast delivery.

- *Accuracy of geocast delivery* [21]: *Accuracy of geocast delivery* is defined as the ratio of the number of group members that actually receive the geocast packet, and the number of group members which were in the geocast region at the time when the geocast delivery was initiated. In our simulation results, we report the average accuracy over all the geocasts performed during the simulation.
- *Overhead of geocast delivery* [21]: The *overhead* is measured in terms of the number of geocast packets *received* by the nodes—the number of geocast packets received by nodes is different from number of geocast packets *sent*, because a single broadcast of a geocast data packet by some node is received by all its neighbors. Specifically, the measures of overhead we use is the *average number of packets* and *average number of bytes* received by each node per geocast. This is calculated by dividing the total number of packets or total number of bytes received by all nodes (over a simulation run) by the number of geocasts performed, and also by the number of nodes in the system. In the pure geocast flooding and LBM scheme, the overhead is due to only data packet, but in GeoTORA it can be due to data as well as control packets.

6.2. Simulation results

In each graph below, one parameter (e.g., pause time, maximum speed, or geocast frequency) was varied while the other parameters were kept constant.

Fig. 8(a) shows the accuracy of geocast delivery of the three geocasting protocols as a function of pause time. As can be expected, pure geocast flooding performs very well, delivering nearly 100% accuracy. LBM shows quite comparable accuracy with pure flooding. Accuracy of GeoTORA is also high, but not as high as pure flooding or LBM. One possible reason for this is that local flooding in GeoTORA may not deliver packets to all nodes in the group. This problem can be solved choosing a larger local flooding



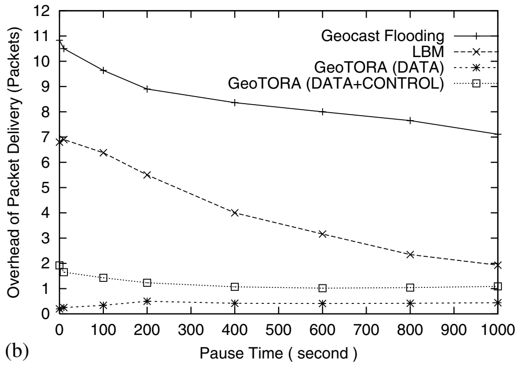


Fig. 8. Comparison of GeoTORA to geocast flooding and LBM protocols with a variation of *pause time* (for 30 nodes, and maximum speed 5.0 units/s): (a) delivery accuracy versus pause time, (b) average number of packets received by a node per geocast versus pause time.

region, trading overhead with accuracy (a similar approach was used in our earlier work on unicast routing [22]).

Delays required to establish a route to the geocast group can be another reason why GeoTORA has lower accuracy than pure flooding. Let us consider the case when the geocast region is empty. With GeoTORA, a source will not send geocast packets until it makes sure a route becomes available. In the meantime, the source will just drop packets, affecting the lower accuracy of geocast packet delivery compared to the other protocols. In contrast, both geocast flooding and LBM protocols allow the source to transmit packets in the same situation, resulting in a higher probability of packet reception by a node just entering the geocast region which was empty.

Fig. 8(b) shows the overhead, i.e., average *number* of geocast packets received by a node per geocast, as a function of pause time. The overhead of GeoTORA consists of data packets as well as control packets (QRY, UPD, and CLR) used to create and maintain routes. The overhead due only to data packets, and overhead due to data and control packets both are plotted separately in the figure.

Generally, the overhead increases with increasing node mobility (i.e., decreasing pause time) for all schemes. However, note that the main reason for increasing overhead in GeoTORA is the control packets, not the data packets. With low mobility rate in GeoTORA, routes for forwarding packets are likely to be fixed and, therefore, the number of control packets to maintain the routes is relatively small. As mobility rate goes up, the cost for a route maintenance process, i.e., number of QRY and UPD packets, also becomes higher.

In Fig. 8(b), the overhead is consistently lower for GeoTORA as compared to geocast flooding and LBM. Recall that GeoTORA limits the scope of flooding to the nodes located in the geocast region. Thus, degree of flooding is smaller in GeoTORA, compared to other two flooding-based protocols. This results in the lower overhead of GeoTORA protocol.

Fig. 9 plots average *amount* (i.e., *Bytes*) of geocast packets received by each node per geocast, with results being similar to Fig. 8(b). Both geocast flooding and LBM have much larger byte overhead than GeoTORA. This means that a larger part of bandwidth is wasted with other two flooding-based protocols. The curve for the GeoTORA byte overhead due to data packets, and the byte overhead due to both data and control packets are almost overlapping because the size of control packets is much smaller than data packets.

Since GeoTORA is based on TORA routing protocol layered on top of Internet Manet Encapsulation Protocol (IMEP) [27], it also uses a neighbor discovery mechanism, which requires each node to transmit at least one *hello* packet per *beacon* period (1 s). This overhead of hello packet transmission should be taken into account separately from that of GeoTORA control packets. We

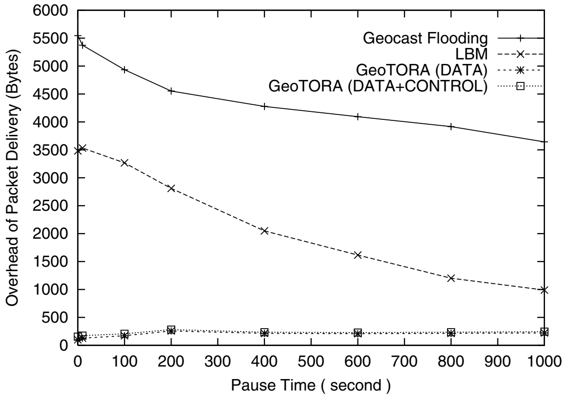


Fig. 9. Average number of bytes versus pause time.

have only looked at the overhead at the IP-level, as in [11,31]. Our decision is based on the observation that the hello packets are also useful for unicast routing, and for any other protocol that might need to detect link failures in the ad hoc networks. Therefore, it is not necessarily fair to attribute hello packet overhead to GeoTORA.

The effect of varying the moving speed of nodes is shown in Fig. 10 in terms of the accuracy and delivery overhead, respectively. Increasing moving speed does not seem to have much impact on the delivery accuracy and overhead of geocast algorithms. In Fig. 10(a), geocast flooding provides the highest accuracy, whereas our GeoTORA shows a slightly lower accuracy than flooding due to reasons discussed previously. However, note that geocast flooding and LBM schemes suffer from a significantly higher overhead (measured in average number of messages per geocast) than GeoTORA for all moving speed (See Fig. 10(b)).

Overhead measured as bytes per geocast (as a function of speed) is also provided in Fig. 11. We can see that GeoTORA performs much better than others.

Finally, in Figs. 12 and 13, we plot accuracy and overhead (number of messages and bytes) of geocast packet delivery with varying geocast frequency. In Fig. 12(b), for GeoTORA, the overhead due to data packets is almost constant. However, as pointed out earlier, GeoTORA's total overhead is due to control packet (QRY, UPD, CLR) and data packets. When geocasts are performed very infrequently, the control overhead of maintaining the DAG becomes high, therefore GeoTORA overhead becomes poor for low geocast frequency.

Now observe Fig. 13 that the overhead of data and control packets measured in *bytes* does not exceed that of LBM for the geocast frequencies simulated. This is unlike Fig. 12(b), where the number of data and control packet does exceed that of LBM. The related overhead is different in the two cases because the size of control packets is much smaller than data packets.

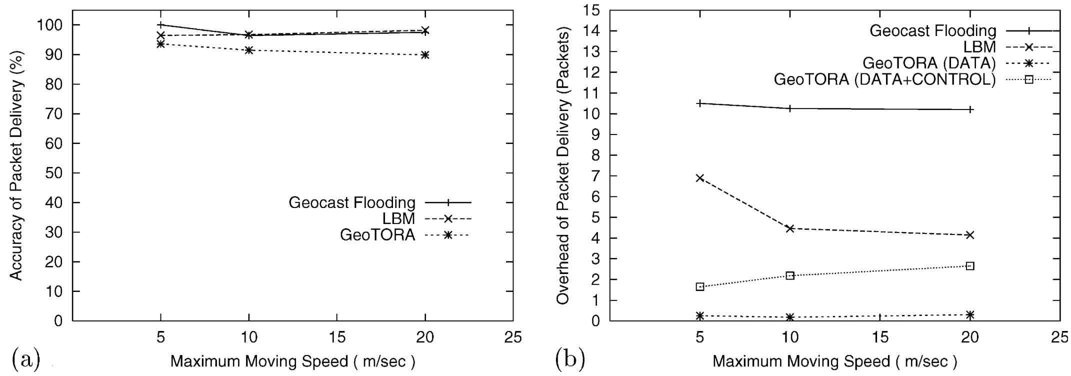


Fig. 10. Comparison of GeoTORA to geocast flooding and LBM protocols with a variation of *speed* (for 30 nodes, and pause time 10 s): (a) delivery accuracy versus moving speed, (b) average number of packets received by a node per geocast versus moving speed.

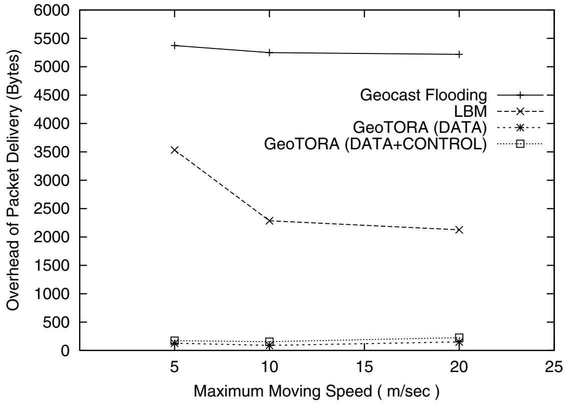


Fig. 11. Average number of bytes versus moving speed.

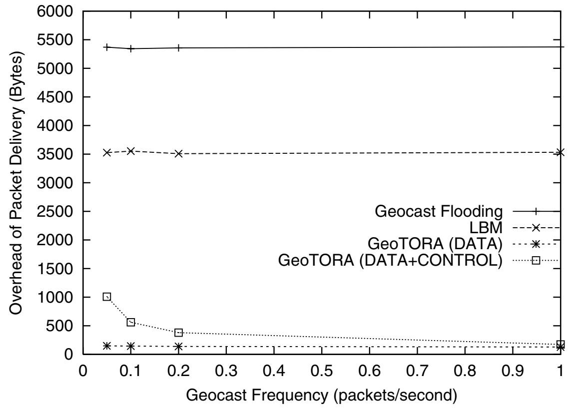


Fig. 13. Average number of bytes versus geocast frequency.

6.3. Impact of hello packets

For all the graphs above, the overhead of hello packet transmission has not been taken into account. As noted earlier, it is not necessarily fair to attribute hello packet overhead to GeoTORA. However, we do so here to provide a conservative upper bound on GeoTORA overhead. Below, we plot the same data for the overhead of geocast delivery from the previous figures, but including the hello packet overhead.

Fig. 14(a) and (b) show the overhead measured as average number of packets per geocast, and the overhead measured as average bytes per geocast, respectively, as a function of pause time. Observe that, for GeoTORA in both figures, one more graph of overhead due to all different types of packets—data, control, and hello—is added to the Figs. 8(b) and 9.

In Fig. 14(a), the average *number* of packets per geocast in GeoTORA is larger than other protocols when the hello packet overhead is also considered as geocast delivery overhead in GeoTORA. This is because at least one hello packet transmission is required per beacon period (i.e., 1 s). For 1000 s simulation with 30 nodes, this results in a minimum overhead of 30,000 packets. However, in Fig. 14(b), the GeoTORA overhead

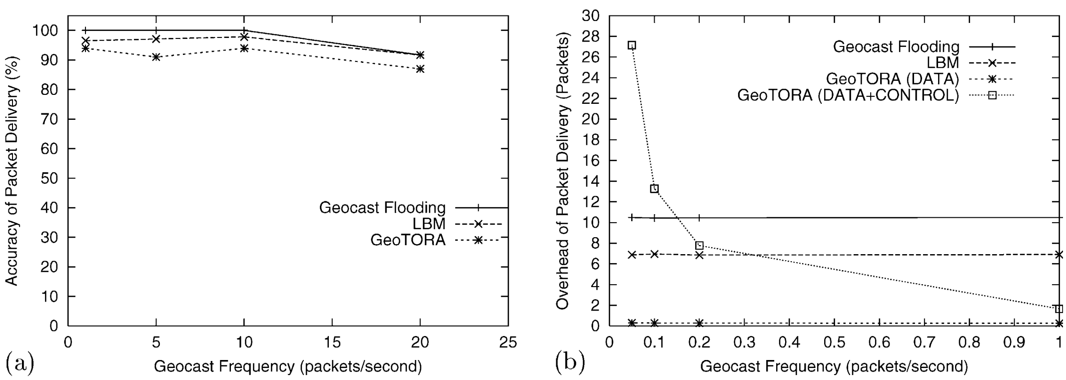


Fig. 12. Comparison of GeoTORA to geocast flooding and LBM protocols with a variation of *geocast frequency* (for 30 nodes, and pause time 10 s): (a) delivery accuracy versus moving speed, (b) average number of packets received by a node per geocast versus geocast frequency.

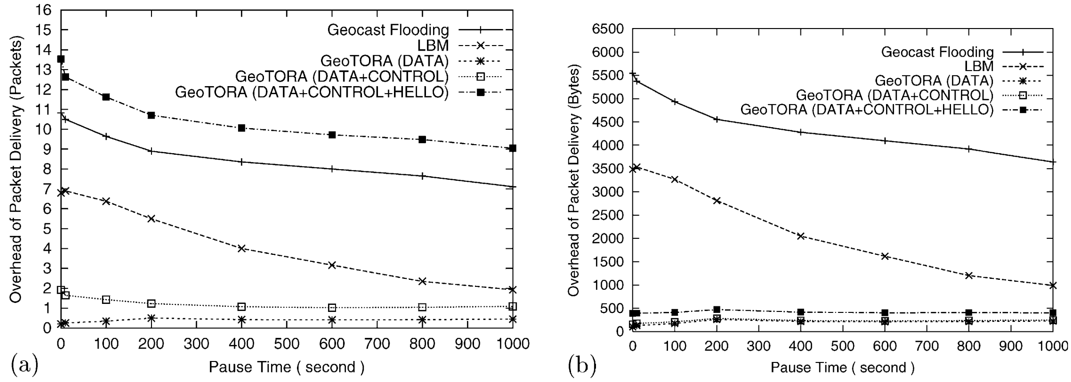


Fig. 14. Overhead comparison of GeoTORA to geocast flooding and LBM protocols with a variation of *pause time* (for 30 nodes, and maximum speed 5.0 units/s): (a) overhead of geocast delivery (packets), (b) overhead of geocast delivery (bytes).

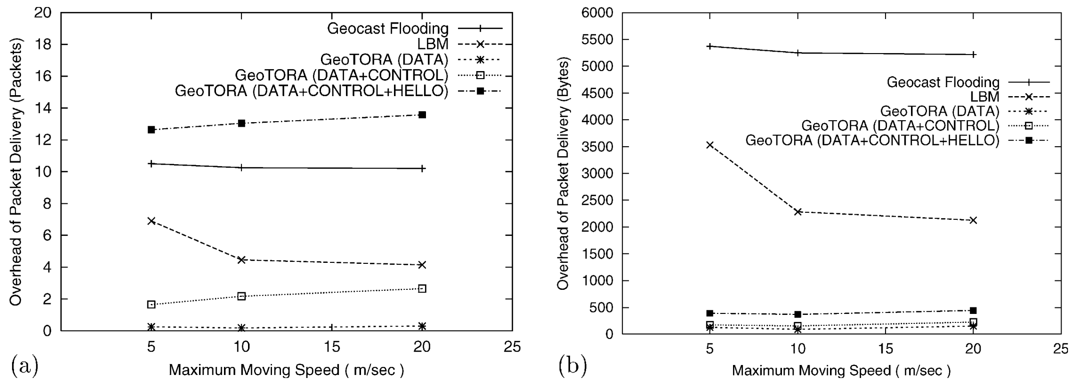


Fig. 15. Overhead comparison of GeoTORA to geocast flooding and LBM protocols with a variation of maximum speed (for 30 nodes, and pause time 10 s): (a) overhead of geocast delivery (packets), (b) overhead of geocast delivery (bytes).

measured as average *bytes* per geocast is still much lower than the geocast flooding and LBM protocols for all the cases—the overhead due only to data, the overhead due to data and control, and the overhead due to data, control and hello packets.

The remaining figures can be explained by the the same reasons above. Fig. 15(a) and (b) present the geocast overhead as a function of moving speed, whereas the effect of varying geocast frequency is shown in Fig. 16(a) and (b) (in terms of the overhead). We can see that, in most scenarios, when the overhead is measured as a total amount of geocast packet delivery, GeoTORA performs much better than other two geocasting protocols (even when hello packets are included in calculating the overhead). However, it is important to observe that the total overhead of GeoTORA including the hello overhead becomes higher than the LBM protocol at low geocast frequencies in Fig. 16(b).

7. Conclusion

We present a novel protocol called GeoTORA for geocasting in mobile ad hoc networks. The basic idea behind GeoTORA is to combine *anycast*

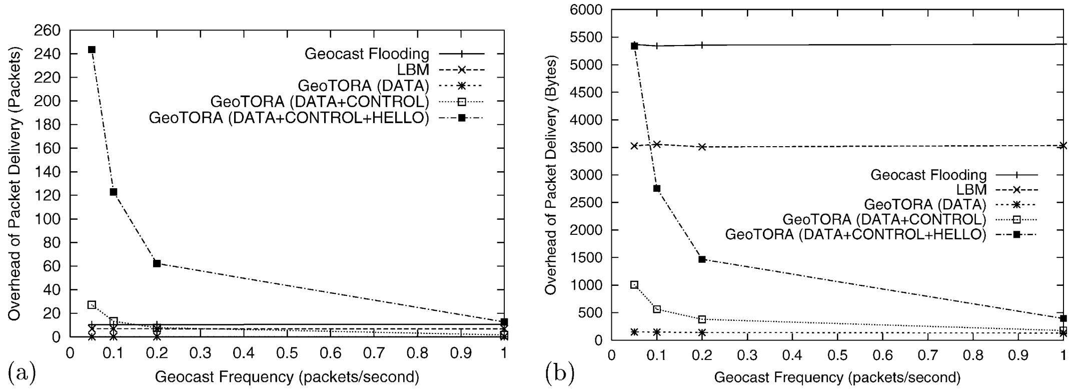


Fig. 16. Overhead comparison of GeoTORA to geocast flooding and LBM protocols with a variation of geocast frequency (for 30 nodes, and pause time 10 s): (a) overhead of geocast delivery (packets), (b) overhead of geocast delivery (bytes).

and *flooding*. In GeoTORA, TORA (unicast) routing protocol has been modified to perform anycast and local flooding has been utilized to limit flood to a small region. As simulation results show, this integration of TORA and local flooding can significantly reduce the geocast message overhead as compared to pure flooding and LBM scheme presented in [21], while achieving high accuracy of geocast delivery. However, note that if geocasts are performed very infrequently then the overhead of GeoTORA can exceed that of LBM.

Acknowledgements

This research is supported in part by National Science Foundation grants ANI-9976626 and ANI-9973152.

References

[1] B. An, S. Tekinay, S. Papavassiliou, A.N. Akansu, A cellular architecture for supporting geocast services, in: Proceedings of the IEEE Vehicular Technology Conference (VTC), Fall 2000, pp. 1452–1459.

[2] M. Bergamo, R.R. Hain, K. Kasera, D. Li, R. Ramanathan, M. Steenstrup, System design specification for mobile multimedia wireless networks (mmwn) (draft), Tech. Rep., BBN Systems and Technologies, October 1996.

[3] J. Boleng, T. Camp, V. Tolety, Mesh-based geocast routing protocols in an ad hoc network, in: Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), San Francisco, CA, April 2001.

[4] E. Bommaiah, A. McAuley, R. Talpade, M. Liu, AMRoute: Ad-hoc multicast routing protocol (Internet-Draft), Mobile Ad-hoc Network (MANET) Working Group, IETF, August 1998.

[5] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Dallas, TX, October 1998, pp. 85–97.

[6] N. Bulusu, J. Heidemann, D. Estrin, Adaptive beacon placement, in: Proceedings of the International Conference on Distributed Computing Systems, Phoenix, AZ, April 2001, pp. 489–498.

[7] S. Capkun, M. Hamdi, J.-P. Hubaux, GPS-free positioning in mobile ad-hoc networks, in: Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS), Hawaii, January 2001, pp. 3481–3490.

[8] C. Chiang, M. Gerla, L. Zhang, Adaptive shared tree multicast in mobile wireless networks, in: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), 1998.

[9] M.S. Corson, A. Ephremides, A distributed routing algorithm for mobile wireless networks, ACM Journal of Wireless Networks 1 (1) (1995) 61–81.

[10] S. Corson, J. Macker, Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations (Internet-Draft), Mobile Ad-hoc Network (MANET) Working Group, IETF, October 1998.

[11] S.R. Das, R. Castaneda, J. Yan, R. Sengupta, Comparative performance evaluation of routing protocols for mobile, ad hoc networks, in: Proceedings of the of IEEE International Conference on Computer Communications

and Networks (IC3N), Lafayette, LA, October 1998, pp. 153–161.
[12] S. Deering, R. Hinden, Internet protocol version 6 (IPv6) specification, RFC 2460, December 1998.
[13] E. Gafni, D. Bertsekas, Distributed algorithms for generating loop-free routes in networks with frequently changing topology, IEEE Transactions on Communications 29 (1) (1981) 11–18.
[14] J.J. Garcia-Luna-Aceves, E.L. Madruga, A multicast routing protocols for ad-hoc networks, in: Proceedings of the IEEE INFOCOM'99, March 1999, pp. 784–792.
[15] Z.J. Haas, A new routing protocol for the reconfigurable wireless networks, in: Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC), San Diego, CA, October 1997, pp. 562–566.
[16] J. Hightower, G. Borriello, Location systems for ubiquitous computing, Computer 34 (8) (2001) 57–66.
[17] C. Ho, K. Obraczka, G. Tsudik, K. Viswanath, Flooding for reliable multicast in multi-hop ad hoc networks, in: The 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL-M'99), August 1999.
[18] T. Imielinski, J.C. Navas, GPS-based geographic addressing, routing, and resource discovery, Communications of the ACM 42 (4) (1999) 86–92.
[19] P. Jacquet, P. Muhlethaler, A. Qayyum, Optimized link state routing protocol (Internet-Draft), Mobile Ad-hoc Network (MANET) Working Group, IETF, February 2000.
[20] D. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publishers, Norwell, MA, 1996.
[21] Y.-B. Ko, N.H. Vaidya, Geocasting in mobile ad hoc networks: Location-based multicast algorithms, in: Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WM-CSA), New Orleans, LA, February 1999, pp. 101–110.
[22] Y.-B. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, Wireless Networks Journal 6 (4) (2000) 307–321.
[23] S. Lee, M. Gerla, C.-C. Chiang, On-demand multicast routing protocol, in: Proceedings of the Wireless Communications and Networking Conference (WCNC), New Orleans, LA, September 1999, pp. 1298–1302.
[24] W.-H. Liao, Y.-C. Tseng, J.-P. Sheu, GeoGRID: A geocasting protocol for mobile ad hoc networks based on GRID, Journal of Internet Technology 1 (2) (2000) 23–32.
[25] W.-H. Liao, Y.-C. Tseng, J.-P. Sheu, GRID: A fully location-aware routing protocol for mobile ad hoc networks, Telecommunication Systems 18 (1) (2001) 37–60.
[26] K. Obraczka, G. Tsudik, Multicast routing issues in ad hoc networks, in: Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC), Florence, Italy, October 1998, pp. 751–756.
[27] V.D. Park, M.S. Corson, An Internet manet encapsulation protocol (IMEP) specification (Internet-Draft), Mobile Ad-hoc Network (MANET) Working Group, IETF, November 1997.
[28] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings of the IEEE INFOCOM'97, Kobe, Japan, April 1997, pp. 1405–1413.
[29] C. Partridge, T. Mendez, W. Milliken, Host anycasting service, RFC 1546, November 1993.
[30] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM), September 1994, pp. 234–244.
[31] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA), New Orleans, LA, February 1999, pp. 90–100.
[32] E.M. Royer, C.E. Perkins, Multicast operation of the ad-hoc on-demand distance vector routing protocol, in: Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), Seattle, WA, August 1999, pp. 207–218.
[33] P. Sinha, R. Sivakumar, V. Bharghavan, MCEDAR: Multicast core extraction distributed ad hoc routing, in: Proceedings of the of Wireless Communications and Networking Conference (WCNC), New Orleans, LA, September 1999, pp. 1313–1317.
[34] C.-K. Toh, A novel distributed routing protocol to support ad-hoc mobile computing, in: Proceedings of the IEEE International Conference on Computing and Communications, March 1996, pp. 480–486.
[35] T. Ozaki, J.B. Kim, T. Suda, Bandwidth-efficient multicast routing for multihop, ad hoc wireless networks, in: Proceedings of the IEEE INFOCOM 2001, Anchorage, AL, April 2000, pp. 1182–1191.
[36] C. Wu, Y. Tay, C.-K. Toh, AMRIS: A multicast protocol for ad hoc wireless networks, in: Proceedings of the MILCOM'99, 1999.

Young-Bae Ko received his Ph.D. in computer science from Texas A&M University, USA. He is currently an Assistant Professor in School of Information and Computer Engineering at Ajou University, South Korea. Prior to joining Ajou University, he worked at IBM T.J. Watson Research Center (New York) as a research staff member. He also worked as a summer intern at Hughes Research Lab (HRL). His research interests are in the areas of mobile computing and wireless networking. In particular, he is actively working on mobile ad hoc networking, wireless local/personal area networks, and pervasive computing. Young-Bae has served on the technical program committees of several conferences and workshops. He also served as the Publicity Co-Chair for the 2001 ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). He is a member of IEEE Computer Society and the ACM.

Nitin Vaidya received Ph.D. from the University of Massachusetts at Amherst. He is presently an Associate Professor of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign (UIUC). He has held visiting positions at Microsoft Research, Sun Microsystems and the Indian Institute of Technology-Bombay. Prior to joining UIUC, he served as an Associate Professor of Computer Science at the Texas A&M University. His current research is in the areas of wireless networking and mobile computing. In particular, he has performed research on routing and medium access control protocols for wireless ad hoc networks, distributed algorithms on ad hoc networks, performance of TCP over wireless networks, and security in wireless networks. His research has been funded by various agencies, including the National Science Foundation, DARPA, BBN Technologies, Microsoft Research and Sun Microsystems. Nitin Vaidya is a recipient of a CAREER award from the National Science Foundation. Nitin has served on the program committees of several conferences and workshops, and served as the General Chair for the 2001 ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc). He serves on the editorial boards of IEEE Transactions on Mobile Computing and IEEE/ACM Transactions on Networking. He is a senior member of IEEE Computer Society and a member of the ACM.